JULY-AUGUST 1976

Technical Supplement

ON RECURSION

THERE EXISTS A BODY OF PROBLEMS WHOSE FORMULATION IS MOST 'NATURALLY' DESCRIBED IN TERMS OF ITSELF, DEFINED AS RECURSIVE PROBLEMS.

IN A RECURSIVE FUNCTION, THE NAME OF THE FUNCTION APPEARS WITHIN THE BODY OF THE SAME FUNCTION ONE OR MORE TIMES, WITH [HOPEFULLY] AN ESCAPE MECHANISM PRESENT FOR EXITTING ONCE THE SOLUTION HAS BEEN FOUND!

A COMMON EXPOSITORY EXAMPLE USED IN MANY APL TEXTS IS THE RECURSIVE DEFINITION OF FACTORIAL N.

WE RECALL THAT N!←→N×(N-1)×(N-2)×....×1

ONE COULD EASILY DEFINE THE ABOVE EXPRESSION RECURSIVELY AS:

```
N!=1           IF N=1

   =N×(N-1)! IF N>1    [HERE WE ARE RESTRICTING OURSELVES TO
                             POSITIVE INTEGERS ≥1]
```

THE RECURSIVE FUNCTION CAN THEN BE DEFINED AS:

```
    ∇ Z←FACT N
[1]    Z←1 ◇ →(1=N)↑0 ◇ Z←N×FACT N-1∇
```

THE FUNCTION CALLS ITSELF TO EVALUATE SUCCESSIVELY SMALLER FACTORIAL N UNTIL N EQUALS 1 AT WHICH POINT IT SURFACES FROM SUCCESSIVE LEVELS OF EXECUTION MULTIPLYING OUT THE TERMS.

FORTUNATELY, OR UNFORTUNATELY, THE NON-RECURSIVE DEFINITION OF THE PROBLEM ALREADY EXISTS AS AN APL PRIMITIVE [!N] WHICH WORKS FOR GENERAL N,SEVERELY CURTAILING THE USEFULNESS OF THE ABOVE EXAMPLE.

AT THE OTHER END OF THE SPECTRUM, WE ENCOUNTER EXTENSIVE APPLICABILITY OF RECURSION IN SYNTAX ANALYSIS WHERE LARGE STRINGS ARE 'DIGESTED' IN RECURSIVELY SMALLER SEGMENTS UNTIL EXECUTABLE EXPRESSIONS ARE FOUND.

FOR EXAMPLE A COMPLEX MATHEMATICAL EXPRESSION LIKE:

(A×X*(B-1)×C)+((D+C)*(E×F))

WOULD BE RECURSIVELY ANALYZED AS

EXPRESSION 1 + EXPRESSION 2

THEN EACH EXPRESSION IN TURN WOULD BE SIMPLIFIED:

EXPRESSION 1 [A×X*(B-1)×C)] =

EXPRESSION 1.1 [A] × EXPRESSION 1.2 [X*(B-1)×C]

THIS PROCESS WOULD CONTINUE UNTIL ALL NESTED EXPRESSIONS HAD BEEN RESOLVED TO THE SIMPLEST CONSTITUENTS [VARIABLES AND MATHEMATICAL EXPRESSIONS +-×*]

A DELIGHTFUL AND SIMPLE EXAMPLE OF A NON-TRIVIAL APPLICATION OF RECURSION IS FOUND IN THE TOWER-OF-HANOI PROBLEM [OR, FINITE ASSIGNMENT ON RANK ORDER PROBLEM]

THE PROBLEM, QUICKLY STATED, IS AS FOLLOWS:

THREE POLES ARE PLANTED IN THE GROUND [SAY, A,B,C], AND N DISKS OF DIFFERENT DIAMETERS ARE PLACED IN THE FIRST POLE SUCH THAT THE SMALLEST IS ON TOP, AND THE LARGEST IS ON THE BOTTOM.
THE OBJECTIVE IS TO MOVE THE DISKS FROM POLE A TO POLE C ACCORDING TO THE FOLLOWING RULES:

1)ONLY ONE DISK CAN BE MOVED AT A TIME

2)A DISK CANNOT BE PLACED ON TOP OF A SMALLER DISK.

```
          A         B         C
          |         |         |
          |         |         |
          -         |         |
         ---        |         |
        -----       |         |
.......................................
```

AFTER ATTEMPTING THIS PROBLEM MANUALLY FOR SMALL N, A PATTERN EMERGES WHICH FORMS THE BASIS OF THE RECURSIVE ALGORITHM:

TO MOVE THE 'BOTTOM' DISK FOR POLE 'A' TO POLE 'C',

(1) *ONE FIRST MOVES ALL BUT THE 'BOTTOM' DISKS TO ONE POLE,*

(2) *THE 'BOTTOM' DISK TO 'C',*

(3) *THEN THE OTHER DISKS ARE PLACED ON TOP OF IT.*

IN THIS CASE 'BOTTOM', 'A', AND 'C' ARE RELATIVE TO THE PARTICULAR NESTING OF THE RECURSIVE FUNCTION.
AS THE OBJECTIVE IS TO HAVE THE PROBLEM COMPUTED A 'LAYER' AT A TIME, EACH CALL TO THE FUNCTION LOOKS AT A SUBSET OF DISKS IN SUCCESSIVELY SMALLER NUMBERS UNTIL NO MORE ARE FOUND, AT WHICH POINT IT RETRACES ITSELF BACK.

THE CORRESPONDING FUNCTION IS AS FOLLOWS:

```
    ∇ Z← N HANOI  PIN
[1] →(N=0)↑0 ◇ (N-1) HANOI PIN[1 3 2]
[2] 'MOVE DISK ';N;' FROM ',PIN[1],' TO ',PIN[3]
[3] (N-1) HANOI PIN [2 1 3]
      ∇
```

LET US EXECUTE THIS FUNCTION WITH 4 DISKS AND PINS 'A', 'B', AND 'C':

```
  4 HANOI 'ABC'
MOVE DISC 1      FROM A TO B
MOVE DISC 2      FROM A TO C
MOVE DISC 1      FROM B TO C
MOVE DISC 3      FROM A TO B
MOVE DISC 1      FROM C TO A
MOVE DISC 2      FROM C TO B
MOVE DISC 1      FROM A TO B
MOVE DISC 4      FROM A TO C
MOVE DISC 1      FROM B TO C
MOVE DISC 2      FROM B TO A
MOVE DISC 1      FROM C TO A
MOVE DISC 3      FROM B TO C
MOVE DISC 1      FROM A TO B
MOVE DISC 2      FROM A TO C
MOVE DISC 1      FROM B TO C
```

AS ANTICIPATED, THE NUMBER OF MOVES WITH N DISKS IS ¯1+2*N.

MOREOVER, THE MAXIMUM LEVEL OF NESTING IS N-1 [*WHY?*].

ASSUMING OUTPUT OF 100 *LINES A MINUTE, ONE ESTIMATES THAT IT WOULD TAKE* 300 *BILLION YEARS TO PRINT THE MOVES FOR* 64 *DISKS* [*SYSTEM SHUTDOWNS NOT WITHSTANDING!*]

THE READER IS ENCOURAGED TO FIND A NON-RECURSIVE DEFINITION OF THE ABOVE PROBLEM.
PLEASE MAIL YOUR ENTRIES [*POSTMARKED NO LATER THAN SEP.* 30] *TO*

HAL CARIM
I.P. SHARP ASSOCIATES,
SUITE 1400,
145 *KING STREET WEST,*
*TORONTO, ONTARIO M*5*H*-1*J*8

FOR THOSE HAVING MAILBOX ACCESS, PLEASE MAILBOX 'HCA' THE WORKSPACE NAME INSTEAD.

NEEDLESS TO SAY PRIZES WILL BE AWARDED...

TO SUMMARIZE, RECURSIVE TECHNIQUES OFTEN OFFER ADVANTAGES IN PROBLEM FORMULATION BY SEGMENTING THE ORIGINAL PROBLEM INTO SMALLER SEGMENTS WHICH ARE SOLVED SIMILARLY; THIS LEADS TO THE FORMULATION OF SIMPLER ALGORITHMS WHICH ARE EASIER TO UNDERSTAND THAN THE NON-RECURSIVE FORM.
ONE OF ITS DRAWBACKS IS THE EXCESSIVE NESTING IN SOME INSTANCES WHICH MIGHT CAUSE WORKSPACE FULL PROBLEMS.